Amendments to the Claims

This listing of claims will replace all prior version, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method for decoding of Reed-Solomon encoded data, comprising the steps of:

receiving a codeword comprising a set of symbols, and calculating a syndrome polynomial $S(x)$ for the received codeword;

receiving erasure information which identifies zero or more symbols in the received codeword that have been declared as a symbol erasure;

calculating a modified syndrome polynomial $T(x)$ from the syndrome polynomial $S(x)$ and then calculating an erasure locator polynomial $\Lambda(x)$, each with reference to the received erasure information;

finding an error locator polynomial $\sigma(x)$ and an errata evaluator polynomial $\omega(x)$, from the modified syndrome polynomial $T(x)$;

determining a location and magnitude of symbol errors and symbol erasures in the received codeword, from the error locator polynomial σ(x), the erasure locator polynomial Λ(x), and the errata evaluator polynomial ω(x); and

correcting the received codeword using the determined location and magnitude of symbol errors and symbol erasures.

2. (Original) The method of claim 1, comprising

receiving the erasure information identifying zero or more of the symbols J as erasures, and calculating a set of terms α^{-v_i} where the set of α^{-v_i} represents locations of the J erasures; and

calculating each of a modified syndrome polynomial T(x) and an erasure locator polynomial Λ(x) using the equation:

$$polyout(x) = polyin(x) \cdot (x+\alpha^{-v_0})(x+\alpha^{-v_1})(x+\alpha^{-v_2}) \cdots (x+\alpha^{-v_{J-1}})$$

by applying polyin(x) an initial value of S(x) to calculate T(x), and applying polyin(x) an initial value of 1 to calculate Λ(x).

3. (Currently amended) The method of claim 1[[or 2]], comprising calculating the modified syndrome polynomial T(x) in a first time multiplexed mode, and then generating the erasure locator polynomial $\Lambda(x)$ in a second time multiplexed mode.

4. (Original) The method of claim 3, comprising calculating the erasure locator polynomial $\Lambda(x)$ in parallel with the step of finding the error locator polynomial $\sigma(x)$ and the errata evaluator polynomial $\omega(x)$.

5. (Currently amended) The method of ~~any of claims 1 to 4~~claim 1, wherein the calculating step comprises calculating each of T(x) and $\Lambda(x)$ using a single polynomial expander.

6. (Currently amended) The method of ~~any preceding claim~~claim 1, wherein finding the error locator polynomial $\alpha(x)$ and the errata evaluator polynomial $\omega(x)$ from the modified syndrome polynomial T(x) comprises solving the key equation:

$$\sigma(x) \cdot T(x) \equiv \omega(x) \bmod x^{2T} .$$

7. (Original) The method of claim 6, comprising solving the key equation by Euclid's algorithm.

8. (Currently amended) The method of ~~any preceding claim~~claim 1, comprising:

finding a location of zero or more symbol errors E by evaluating the error locator polynomial σ(x) such that if σ(x) = 0 for some $x = \alpha^{-i}$ then an error has occurred in symbol *i*, and evaluating a derivative σ'(x) of the error locator polynomial σ(x);

finding a location of zero or more symbol erasures J by evaluating the erasure locator polynomial Λ(x) such that if Λ(x) = 0 for some $x = \alpha^{-i}$ then an erasure has occurred in symbol *i*, and evaluating a derivative Λ'(x) of the erasure locator polynomial Λ(x);

evaluating the errata evaluator polynomial ω(x); and

determining an error magnitude for each symbol error by solving the equation:

$$E_i = \frac{\omega(x)}{\sigma'(x) \cdot \Lambda(x)} \text{ for } x = \alpha^{-i}; \text{ and}$$

determining an erasure magnitude for each symbol erasure by solving the equation:

$$J_i = \frac{\omega(x)}{\sigma(x) \cdot \Lambda'(x)} \text{ for } x = \alpha^{-i}.$$

9. (Currently amended) The method of ~~any preceding claim~~claim 1, comprising:

transforming the error locator polynomial σ(x), the erasure locator polynomial Λ(x), and the errata evaluator polynomial ω(x) such that each coefficient i is transformed by a factor of $\alpha^{(2^W-B)i}$, where GF(2^W) is the Galois field of the Reed Solomon code used to generate the received codeword and B is a number of symbols in the received codeword.

10. (Original) A Reed-Solomon decoder, comprising:

a syndrome block arranged to calculate a syndrome polynomial S(x) from a received codeword;

an erasurelist block for receiving erasure information which identifies zero or more symbols in the received codeword as symbol erasures;

a polynomial expander arranged to calculate a modified syndrome polynomial T(x) from the syndrome polynomial S(X) and arranged to calculate an erasure locator polynomial Λ(x), each with reference to the erasure information;

a key equation block arranged to find an error locator polynomial σ(x) and an errata evaluator polynomial ω(x), from the modified syndrome polynomial T(x);

a polynomial evaluator block and a Forney block arranged to determine a location and magnitude of symbol errors and symbol erasures in the received codeword, from to the error locator polynomial $\sigma(x)$, the erasure locator polynomial $\Lambda(x)$, and the errata evaluator polynomial $\omega(x)$; and

a correction block arranged to correct the received codeword from the determined location and magnitude of each symbol error and each symbol erasure.

11. (Original) The decoder of claim 10, wherein the polynomial expander is time multiplexed between a first mode for generating $T(x)$, and a second mode for generating $\Lambda(x)$.

12. (Original) The decoder of claim 11, wherein the polynomial expander operates in the second mode to calculate the erasure locator polynomial $\Lambda(x)$ in parallel with the key equation block finding an error locator polynomial $\sigma(x)$ and an errata evaluator polynomial $\omega(x)$.

13. (Currently amended) The decoder of claim 10, [[11 or 12,]] comprising:

a first polynomial evaluator arranged to find a location of zero or more symbol errors E by evaluating the error locator polynomial $\sigma(x)$ such that if $\sigma(x)=0$ for some $x = \alpha^{-i}$ then an error has occurred in symbol i;

a second polynomial evaluator arranged to find a location of zero or more symbol erasures J by evaluating the erasure locator polynomial $\Lambda(x)$ such that if $\Lambda(x)=0$ for some $x = \alpha^{-i}$ then an erasure has occurred in symbol i;

the first and second polynomial evaluators being arranged to evaluate a derivative $\sigma'(x)$ of the error locator polynomial $\sigma(x)$, and a derivative $\Lambda'(x)$ of the erasure locator polynomial $\Lambda(x)$, respectively;

a third polynomial evaluator arranged to evaluate the errata evaluator polynomial $\omega(x)$; and

a Forney block arranged to determine an error magnitude for each symbol error E by solving the equation

$$E_i = \frac{\omega(x)}{\sigma'(x)\cdot\Lambda(x)} \text{ for } x = \alpha^{-i}, \text{ and}$$

determining an erasure magnitude for each symbol erasure J by solving the equation

$$J_i = \frac{\omega(x)}{\sigma(x)\cdot\Lambda'(x)} \text{for } x = \alpha^{-i}.$$

14. (Currently amended) The decoder of ~~any of claims 10 to 13 claim~~claim 10, comprising:

a transform block arranged to transform each of the error locator polynomial $\sigma(x)$, the erasure locator polynomial $\Lambda(x)$, and the errata evaluator polynomial $\omega(x)$ such that each coefficient i is transformed by a factor of $\alpha^{(2^W-B)i}$, where $GF(2^W)$ is the Galois field of the Reed Solomon code used to generate the received codeword and B is a number of symbols in the received codeword.

15. (Original) A method for use in decoding Reed-Solomon encoded data, comprising:

receiving a codeword comprising a set of symbols, and calculating a syndrome polynomial S(x) from the received codeword;

receiving erasure information identifying zero or more of the symbols as J erasures, and calculating a set of terms α^{-v_i} where the set of α^{-v_i} represents locations of the J erasures; and

calculating each of a modified syndrome polynomial T(x) and an erasure locator polynomial $\Lambda(x)$ using the equation:

$$polyout(x) = polyin(x)\cdot(x+\alpha^{-v_0})(x+\alpha^{-v_1})(x+\alpha^{-v_2})\cdots(x+\alpha^{-v_{J-1}})$$

by applying polyin(x) an initial value of S(x) to calculate T(x), and applying polyin(x) an initial value of 1 to calculate $\Lambda(x)$.

16. (Original) A Reed-Solomon decoder comprising:

a syndrome calculation block arranged to receive a codeword comprising a set of symbols, and calculate a syndrome polynomial S(x) from the received codeword;

an erasure list block arranged to receive erasure information identifying zero or more of the symbols J as erasures, and calculate a set of terms α^{-v_i} where the set of α^{-v_i} represents locations of the J erasures; and

a polynomial expander block arranged to calculate each of a modified syndrome polynomial T(x) and an erasure locator polynomial Λ(x) using the equation:

$$polyout(x) = polyin(x) \cdot (x+\alpha^{-v_0})(x+\alpha^{-v_1})(x+\alpha^{-v_2}) \cdots (x+\alpha^{-v_{J-1}})$$

by applying polyin(x) an initial value of S(x) to calculate T(x), and applying polyin(x) an initial value of 1 to calculate Λ(x).

Claims 17-18 (Canceled)